Pipeline Pilot Interface to *FTrees*

Version 2.4.5.1

User Guide

(for *FTrees* version 2.4.5 and above and Pipeline Pilot version 8.0 and above)

Edgar Derksen, Sally Hindle

Fraunhofer Institut Algorithmen und Wissenschaftliches Rechnen

The idea of Feature Trees was born in 1997 during Matthias Rarey's six-month research stay at SmithKline Beecham Pharmaceuticals R&D, King of Prussia, PA (USA) and then further developed at Institute for Algorithms and Scientific Computing (SCAI), then part of the German National Research Center for Information Technology (GMD) and now the Fraunhofer Gesellschaft (FhG). Since 2002 BioSolveIT GmbH has been responsible for the licensing and continuing development of the *FTrees* software.

At this point we would like to thank Scott Dixon (SmithKline Beecham, now Metaphorics LCC.), Markus Wagener (SmithKline Beecham, now N.V. Organon), and Jens Lösel for a lot of helpful and constructive discussions during Matthias Rarey's stay at Smithkline Beecham and afterwards. Without them, the idea of Feature Trees would not have been evolved in the way it has. Also, Matthias Rarey thanks the GMD and Smithkline Beecham for funding his research stay in King of Prussia.

In summer 2000, the Feature Tree comparison algorithms were extended to search directly in large combinatorial chemistry spaces. A two-stage dynamic programming algorithm enables searching directly in chemistry spaces without an explicit enumeration of molecules [2]. This work was also done during a research stay in the US, this time at Roche Bioscience in Palo Alto. The chemistry space search algorithm was developed in cooperation with Martin Stahl (Hoffmann-La Roche, Basel) and we would like to thank him for this excellent cooperation. We also wish to thank Hans-Joachim Böhm, Hans Maag (both Roche), and Thomas Lengauer (GMD) for making this research stay possible.

Since then the Feature Trees software has been further developed and extended by several contributors including Marc Zimmermann (FhG - MTrees and the new Dynamic Matchsearch algorithm), Robert Fischer, Sally Hindle, and other developers at BioSolveIT GmbH and the Center for Bioinformatics (ZBH), University of Hamburg.

This document contains proprietary information of BioSolveIT GmbH and is protected by copyright. It is provided together with Software of BioSolveIT under a license agreement and may be used only in accordance with the terms and conditions of this agreement. The document serves solely for the purpose of using the Software. No part of the document may be transferred to any third party or reproduced as a whole or in parts without written permission from BioSolveIT.

Base software: © 2001 by Fraunhofer Gesellschaft (FhI-SCAI); Getline library: © 1993 by Chris Thewalt; PVM library: © 1997 by University of Tennessee, Knoxville TN; Python library: © 1991-1995 by Stichting Mathematisch Centrum, Amsterdam, The Netherlands.

© 2015 BioSolveIT GmbH, An der Ziegelei 79, 53757 St. Augustin, Germany
Phone ++49-2241-2525-0, support@biosolveit.de

Contents

Contents **3**

1 Quick Start Steps **5**

1.1 Download and Import current *FTrees* Package 5

1.2 Install *FTrees* 5

1.3 Create or Update Protocols 5

2 Introduction **7**

2.1 About *FTrees* 7

2.2 *FTrees* Components in Pipeline Pilot 7

2.2.1 *FTrees* Calculator 7

2.2.2 *FTrees* Writer 8

2.2.3 *FTrees* Reader 8

2.2.4 *FTrees* Similarity 8

2.3 Recommended protocols when using *FTrees* in Pipeline Pilot 8

2.3.1 Load Molecules, Calculate Feature Trees and Store a Dataset 8

2.3.2 Load Molecules with Feature Trees and Compare to Query Molecules 9

2.3.3 Calculate Feature Trees on-the-fly and Compare to Query Molecules . 10

2.3.4 The Advantages and Disadvantages of the Various Storage Formats . 11

3 Installation **13**

3.1 BioSolveIT Web Installation 13

3.2 Using the *FTrees* Installation Component 13

3.3 Using Global Variables 14

3.4 Using Custom External *FTrees* Installations 14

3.4.1 To Connect to a *FTrees* Installation on the Pipeline Pilot Server 15

3.4.2 To Connect to a *FTrees* Installation on a Remote Linux Server 16
3.5 Running *FTrees* in Parallel in Pipeline Pilot 17
3.5.1 The Default Set-Up: Maintaining the Pipeline Effect 18
3.5.2 Running in Parallel on the Pipeline Pilot Server 19
3.5.3 Running in Parallel on a Remote Linux Cluster 19
3.5.4 Example Scenarios and Required Settings 22
3.6 Uninstallation 23

4 Trouble Shooting **25**
4.1 Problems connecting to the external installation 25
4.2 Problems using the *ssh* Method 26
4.3 Further help and BioSolveIT PDF Reporter 26

5 Tips and Tricks **29**
5.1 Other Significant Parameters in the *FTrees* Components 29
5.1.1 <Molecule Initialization> 29
5.1.2 `.ftdb` Compression 30
5.1.3 Fail ports for *FTrees* Calculator and Simlarity 30
5.1.4 <File Format> 30
5.1.5 <Has Same File System> 31
5.2 Writing a standard *FTrees* `.fdf` file 31
5.3 Reading a standard *FTrees* `.fdf` file for the Similarity Component 31
5.4 Accessing Other Domains within Pipeline Pilot 31

References **33**

Bibliography **33**

1

Quick Start Steps

1.1 Download and Import current *FTrees* Package

1. Download current "*FTrees* in PipelinePilot" package on
   http://www.biosolveit.de/download/
2. Extract *FTrees* package to a custom directory.
3. From the custom directory import all components (.xml) using PP-Client via drag and drop to your components collection.

1.2 Install *FTrees*

1. Drag and drop *FTrees* installation component from the components collection to an empty protocol.
2. Add your license for the executable of *FTrees* in the field for the parameter <Run FTrees -> on PP Server -> License Server or License File> in the Implementation tab – if your license is available from a license server, simply type in the name of the server in this format *@servername*, or if you have a license file then you may browse for it using the ... facility.
3. Run protocol once, it will result into a html report showing installation success.
4. For alternative installations and more details see chapter 3.

1.3 Create or Update Protocols

1. Drag and drop new components into a new protocol.
2. Update via a *right click drag and drop* of new component onto component reference in a protocol.
3. Now you are ready to setup the new components and run your protocol.

2

Introduction

2.1 About *FTrees*

FTrees is a piece of software for calculating the Feature Tree descriptor and comparing two or more of these descriptors to each other. The theory behind Feature Trees can be found in [1].

Rather than being based on a linear description such as bit strings or vectors, the Feature Tree descriptor represents the molecule as an unrooted tree where the nodes of the tree describe the major building blocks of the molecule. The comparison of two Feature Trees then proceeds using a recursive matching algorithm, splitting the trees into smaller and smaller subtrees. The Feature Tree approach has several advantages, the most important being the fact that the alignment of two Feature Trees can be translated into a comprehensible mapping of the two underlying molecules. For more details on the algorithms and achieved results see [1].

2.2 *FTrees* Components in Pipeline Pilot

We have designed several different components based around the *FTrees* software to enable you to integrate *FTrees* into your various pipelines with total flexibility, while maintaining a speedy protocol. As with other similarity calculator components in Pipeline Pilot, the *FTrees* components allow you to compare the molecules in the pipeline to given queries and and optionally apply a filter so that only similar molecules pass on to the next components of the pipeline. It is also important to note that the key to a speedy comparison is the fact that you do not need to calculate a Feature Tree descriptor every time for your pipeline molecules – in this version of *FTrees* for Pipeline Pilot we offer you various possibilities for storing and re-using the Feature Trees descriptor in a Pipeline Pilot consistent way.

The components are:

2.2.1 *FTrees* Calculator

Calculates the Feature Tree descriptor for the incoming molecule. If the descriptor generation was successful, the molecule is output to the "Pass" port – the Feature Tree can be seen

as a new property in the molecule. Molecules for which no Feature Tree could be calculated are output to the "Fail" port. Molecules that already have a Feature Tree attached to them skip the component and are diverted directly to the "Pass" port.

2.2.2 *FTrees* Writer

Molecules in a Pipeline Pilot pipeline that already have a Feature Tree can be written to a special database designed for use with the *FTrees* Pipeline Pilot components. The molecules are stored complete with all data in a compromised format. The resulting database is a file stored in a user-defined location on the file system. Instead of *FTrees* Writer/Reader you may substitute instead the Pipeline Pilot Cache Writer/Reader or the SD Writer/Reader.

2.2.3 *FTrees* Reader

This component reads the special *FTrees* database written by the *FTrees* Writer component.

2.2.4 *FTrees* Similarity

This is the central *FTrees* component. A query molecule file is defined as a parameter to this component. Every incoming pipeline molecule that has a Feature Tree is compared to each of the queries and receives a similarity score for each. If a threshold is defined, only molecules with a similarity greater than the threshold are output to the "Pass" port. If the comparison failed (for example, the molecule had no Feature Tree), or if the similarity is less than a defined filter threshold the molecules are passed to the "Fail" port.

2.3 Recommended protocols when using *FTrees* in Pipeline Pilot

Comparing Feature Trees to calculate a similarity value is *not* the time consuming step when using *FTrees*. However, the actual calculation of the Feature Tree descriptor can be quite expensive timewise. Therefore, we recommend that you always save a dataset of molecules for re-use once you have calculated the Feature Tree descriptors for them. There are several ways to do this but of course it is not compulsory to do so. Here are a couple of suggested protocols we would recommend you use to get the most speedy results with *FTrees* in Pipeline Pilot.

2.3.1 Load Molecules, Calculate Feature Trees and Store a Dataset

1. Load molecules into the pipeline
2. Optionally filter, create properties etc. – process your molecules
3. Use the *FTrees* Calculator component to create Feature Trees for the molecules

Figure 2.1: Use the *FTrees* Calculator component to create feature trees for molecules in the pipeline. Save them for re-use with the *FTrees* Writer

4. Optionally further process the molecules
5. Use *FTrees* Writer to write the dataset of molecules to the special *FTrees* in Pipeline Pilot database file format. Alternatively, you may store the molecules with the Feature Tree descriptor using the Cache Writer or in SD molecule format using the SD Writer. You may see the advantages and disadvantages of each format listed in the table below.

See 2.1 for an example.

2.3.2 Load Molecules with Feature Trees and Compare to Query Molecules

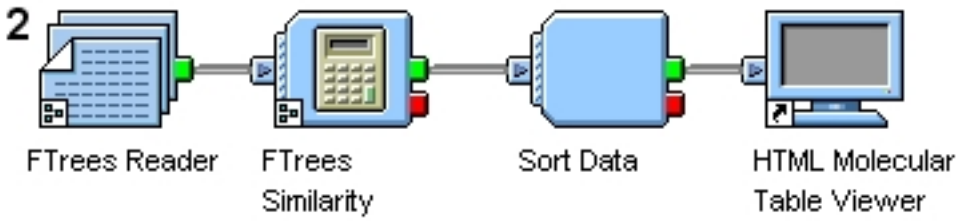

Figure 2.2: Load molecules containing feature trees into the pipeline using the *FTrees* Reader. Calculate the similarity to query molecules from file with the *FTrees* Similarity component.

1. Use the *FTrees* Reader to load molecules from the special *FTrees* in Pipeline Pilot database file format. Alternatively, if you stored the molecules to a Cache or as SD format, use the appropriate Reader.
2. Optionally process the molecules
3. Use the *FTrees* Similarity component to compare the pipeline molecules to query molecules found in a defined molecule file. The queries can be in a molecule file, Feature Tree file (`.fdf` format), or the special *FTrees* in Pipeline Pilot database file format. Optionally, add a threshold similarity value to filter the molecules that exit the "Pass" port.
4. . . . continue your pipeline to sort, further filter and process the molecules

See 2.2 for an example.

2.3.3 Calculate Feature Trees on-the-fly and Compare to Query Molecules

This is a much slower way to compare your molecules than the recommended protocol in section 2.3.2. However, it gives you full flexibility as the descriptors must always be calculated at least once at some time or other! Adding a Writer component further down the pipeline still means you can save the dataset for further use later.

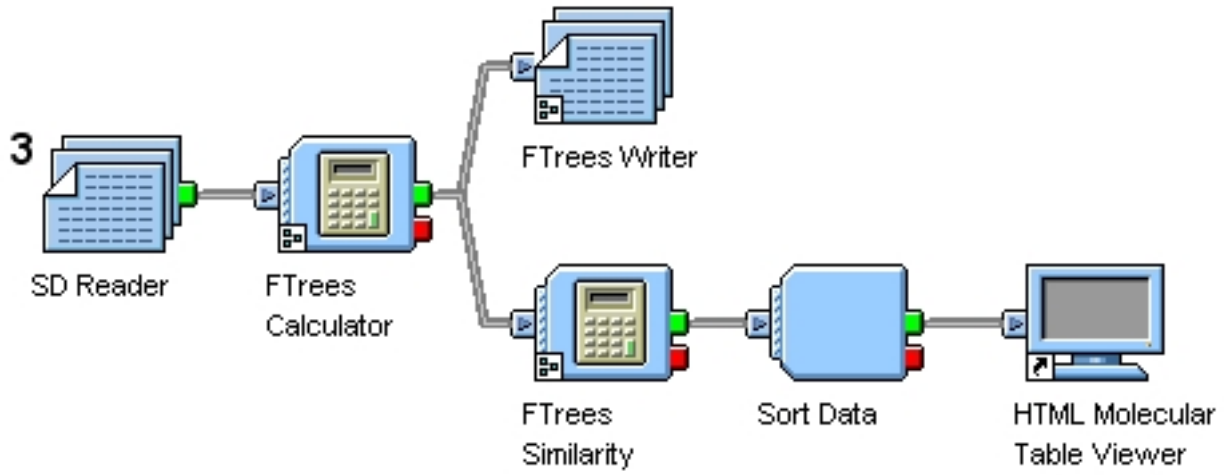

Figure 2.3: Calculate feature trees for the pipeline molecules and then the similarity directly. Remember though to save your molecules with feature trees for re-use as the *FTrees* Calculator is the time-consuming component, not the *FTrees* Similarity component.

1. Load molecules into the pipeline
2. Optionally process the molecules
3. Use the *FTrees* Calculator component to create Feature Trees for the molecules
4. Optionally process the molecules
5. Use the *FTrees* Similarity component to compare the pipeline molecules to query molecules found in a defined molecule file. The queries can be in a molecule file, Feature Tree file (`.fdf` format), or the special *FTrees* in Pipeline Pilot database file format. Optionally, add a threshold similarity value to filter the molecules that exit the "Pass" port.
6. Optionally process the molecules
7. Use *FTrees* Writer to write the dataset of molecules to the special *FTrees* in Pipeline Pilot database file format. Alternatively, you may store the molecules with the Feature Tree descriptor using the Cache Writer or in SD molecule format using the SD Writer. You may see the advantages and disadvantages of each format listed in the table below.

See 2.3 for an example.

2.3.4 The Advantages and Disadvantages of the Various Storage Formats

The values given here are for processing a typical small molecule dataset containing just over 33 000 molecules.

Times are those simply reported in the Pipeline Pilot GUI – we mean here only to compare the relative time of each process.

"User defined file location" means the user enters the location where the file should be stored on disk. The file is then independent of Pipeline Pilot and can be moved or deleted etc. outside of Pipeline Pilot. The Cache is a special Pipeline Pilot internal storage type – note this quote from the Pipeline Pilot documentation:

Since the caches created with scopes User Only *and* All Users *may be accessed by multiple jobs, you should use caution as problems may arise if two jobs try to write to the same shared cache at once. Also, to save disk space, you are responsible for deleting these caches when finished, using the Delete Cache component.*

	Reader/Writer type:		
	FTrees	Cache	SD
Write speed (secs)	26	6	3
File size (MB)	58	36	113
Read speed (secs)	17	5	3
User defined file location	yes	no	yes
ASCII format file	no	no	yes

3

Installation

3.1 BioSolveIT Web Installation

The easiest way to download and install BioSolveIT packages and tools is to download and run BioSolveIT Web Installer component. Download BioSolveIT in PipelinePilot package from http://www.biosolveit.de/download. Read UserGuide within that package for further details.

Note that both PP-Client and PP-Server need web access to run BioSolveIT Web Installer. Without web access you need to follow one of the following sections in this chapter.

3.2 Using the *FTrees* Installation Component

The *FTrees* components are set by default to use a so-called auto installation of *FTrees*. This works as follows. When you run either of the *FTrees* Calculator or Similarity components (and the parameter <Run FTrees> is set to *on PP Server* and the parameter <Run FTrees-> on PP Server -> Use> is set to *FTrees Auto Installation*), they search for a *FTrees* installation in the directory:

```
<scitegic install directory>/public/bin/BioSolveIT/
```

and use this installation to run the calculation. A html report pupup shows if a *FTrees* installation is not found in this directory.

Use *FTrees* installer component to install *FTrees* prior to run calculations. In case of a default setup you don't need administrator rights to do this as it is actually Pipeline Pilot that carries out the installation and not you as a user. In special cases your system administrator prevents installations by PP and you need to ask him to install *FTrees* .

As this installation knows nothing about the licenses you may have for *FTrees*, you have to supply the license information separately. This is supplied using the parameter <Run FTrees -> on PP Server -> License Server or License File> as described above in the section 1.

The installation is carried out once only (for that *FTrees* version) and only once per server not per user. The software is available to all users once it is installed.

3.3 Using Global Variables

Use global variables as an alternative to the *FTrees* installation component decribed in section 3.2. Note that global variables will only be used if components are set to run auto installation (see. section 3.2) but no installation is done by an *FTrees* installation component yet.

Goto Administratin Portal accessible through the Server Home Page at http://localhost:9944/ if PP-Server is installed on your local machine. The Server Home Page is also available through your PP-Client in menu <Help -> Server Home Page>. Use default name: scitegicadmin and password: scitegic if you have not changed it yet. Browse to <Setup -> Globals>. Add Custom Global Protocol Properties.

Note that defining parameters within your protocol will override global variables if they are named the same. Use parameters within your protocol instead of global variables only if you want to use a custom installation for that protocol.

The first variable file_server_*FTrees*_executable: must be set to the full path of *FTrees* executable available by the PP-Server. Example:

```
C:\Program Files (x86)\BioSolveIT\FTrees\ftrees.exe
```

The second variable file_server_*FTrees*_config must be set to the full path of *FTrees* configuration. Example:

```
C:\Program Files (x86)\BioSolveIT\FTrees\config_ft.dat
```

In comparison to *FTrees* auto installation (see section 3.2) you need to check the compatibility between your defined installation and the components.

3.4 Using Custom External *FTrees* Installations

You can also use a custom external installation of the *FTrees* software at each component. This means you also have the opportunity to use settings different to those set by default. To do this, you must already have *FTrees* installed somewhere on your system outside of Pipeline Pilot. To install *FTrees* yourself, visit the download page at BioSolveIT:
`http://www.biosolveit.de/download`
and fetch the download package for your system for the latest *FTrees* package. Follow the instructions in the package to install *FTrees* and receive your licenses. Enter the license information for *FTrees* as described in the package **and not using the parameter <Run FTrees -> on PP Server -> License Server or License File> as for the internal installation**.

To use an external installation of *FTrees*, you must change the value of the parameter <Run FTrees-> on PP Server -> Use> in the Implementation tab to *preinstalled FTrees*.

There are actually two ways to use *FTrees* with an external installation. These are by using *FTrees* installed directly on the Pipeline Pilot server , or by accessing a remote machine where *FTrees* is installed using *ssh* method. The method is selected using the parameter <Run FTrees> Both methods are covered in more detail below.

3.4.1 To Connect to a *FTrees* Installation on the Pipeline Pilot Server

The most common scenario is that you will have an installation of *FTrees* on the Pipeline Pilot server. If you choose this option, you just have to enter the path to the executable and configuration file as parameters in the Implementation tab. Pipeline Pilot will then just start *FTrees* whenever it is required by making a call to the executable that you entered.

This method is given the name *on PP Server* – On Pipelinepilot Server. You can see an example in figure 3.1.

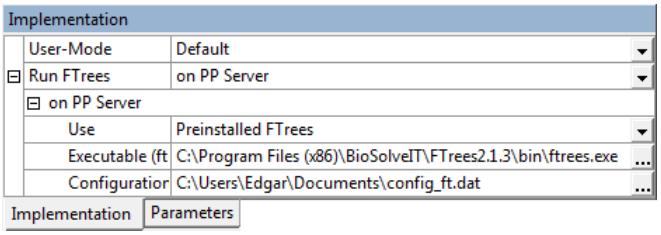

Figure 3.1: *on PP Server*: Connect to an *FTrees* Installation on the Pipeline Pilot Server

- Requirement:
  - *FTrees* is installed on the Pipeline Pilot server. You can see which machine is the Pipeline Pilot server by starting your copy of Pipeline Pilot Client on your own workstation and find the name or IP of the server shown at the bottom right of the status bar (see figure 3.2). You must find where the *FTrees* installation is *on that machine*.

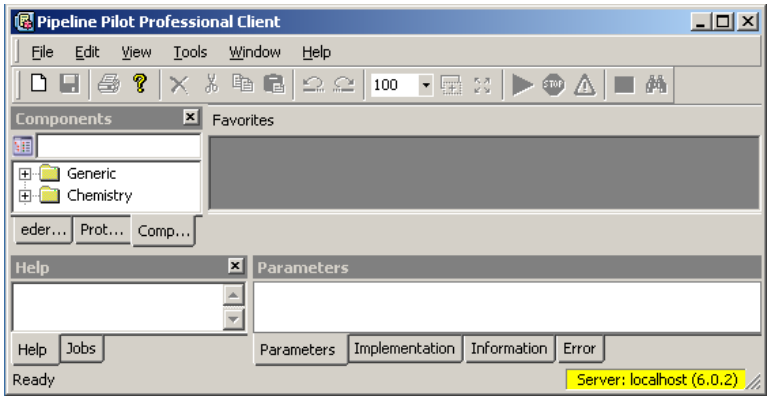

Figure 3.2: See where your Pipeline Pilot Server is installed.

- General Steps:
  1. Set <Run FTrees> to *on PP Server*
  2. Set <Run FTrees-> Use> to *Preinstalled FTrees*

3. For the parameter <Run FTrees-> on PP Server -> Executable>, enter the path to the *FTrees* installation on Pipeline Pilot server. For example:
   `C:\Programs\BioSolveIT\FTrees2\bin\ftrees.exe`
4. For the parameter <Run FTrees-> on PP Server -> Configuration>, enter the path to the *FTrees* configuration file config_ft.datassociated with the *FTrees* installation. For example:
   `C:\Programs\BioSolveIT\FTrees2\config_ft.dat`

You can save these settings in the components.

3.4.2 To Connect to a *FTrees* Installation on a Remote Linux Server

Your existing *FTrees* installation could be on a Linux computer remote from the Pipeline Pilot server – in this case we offer an alternative so you can use the remote installation instead. Here, the calculations carried out by *FTrees* will be done on the remote Linux machine. Pipeline Pilot logs into the Linux machine using `ssh`, having copied all relevant files to the machine, and will run the calculation there, finally it needs copying back all data to the Pipeline Pilot server to continue with the pipeline.

This method is given the name *via SSH*. You can see an example in figure 3.3.

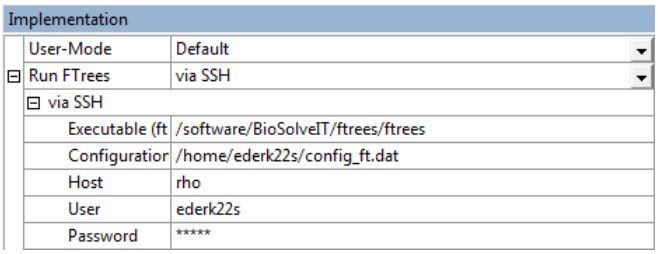

Figure 3.3: *via SSH*: Connect to an *FTrees* Installation on a Remote Linux Server via ssh

- Requirement:
  - *FTrees* is installed on a Linux machine available to the Pipeline Pilot server via `ssh`.
- General Steps:
  1. Set <Run FTrees> to *via SSH*
  2. For the parameter <Run FTrees-> via SSH -> Executable>, enter the path to the *FTrees* installation on the Linux machine. For example:
     `/software/BioSolveIT/ftrees-2.0.1/bin/ftrees-2.0.1`
  3. For the parameter <Run FTrees-> via SSH -> Configuration>, enter the path to the *FTrees* configuration file config_ft.datassociated with the *FTrees* installation. For example:
     `/software/BioSolveIT/ftrees-2.0.1/config_ft.dat`

4. For the parameter <Run FTrees-> via SSH -> Host>, enter the Linux machine host name

- User specific steps:

1. For the parameter <Run FTrees-> via SSH -> User>, enter the user login name for ssh on the Linux machine
2. For the parameter <Run FTrees-> via SSH -> Password>, enter the user password for ssh on the Linux machine
3. There are more advanced options to be found under <Run FTrees-> via SSH -> Options> for more specific ssh parameters. Note the option <Run FTrees-> via SSH -> Options -> Delete Results> may be useful for trouble-shooting later.

You can save these settings in the components (be sure not to save your own user specific login details in components available to others!).

All files necessary for the *FTrees* calculation will be transferred via `scp` between the Pipeline Pilot server and ssh Linux machine. Files copied and files created the remote server are automatically deleted at the end of the job leaving no trace. However, in case the user would like to leave a copy of the calculation and result files on the Linux machine, or for trouble-shooting as mentioned above, it is possible to set a parameter to tell Pipeline Pilot not to delete these files:
<Run FTrees-> via SSH -> Options -> Delete Results> : *False*

3.5 Running *FTrees* in Parallel in Pipeline Pilot

FTrees in Pipeline Pilot takes advantage of the parallel computing options available in Pipeline Pilot to speed up longer calculations. This section tells you how to adjust the options to your system and needs. These options are only relevant to the *FTrees* Calculator and Similarity components. You will find the options in the Implementation tab, as in figure 3.4.

The most important limitation to a parallel processing calculation is the number of *FTrees* licenses that you have. If you only have a single license then parallel calculations will not be possible. Further choices in the set-up of the parallel computing computing calculation depend on the number of Pipeline Pilot licenses you have plus your choice of connection to the external *FTrees* installation (*on PP Server* or *via SSH*).

It is important to note that a balance must be achieved between the overhead caused by running several calculations instead of one and the size of the calculation – there is a lot of overhead involved in sending all the essential data to different computers and collecting the results. For the *FTrees* Similarity component we think parallel computing will only benefit very large jobs, above 200 000 pipeline molecules or so. For the *FTrees* Calculator this number is much lower, as the calculation of feature trees is much slower. We suggest 5 000. Of course, these figures depend on the speed of your machines and network: experiment with your set-up if you intend to carry out such large calculations often. Also we advise you

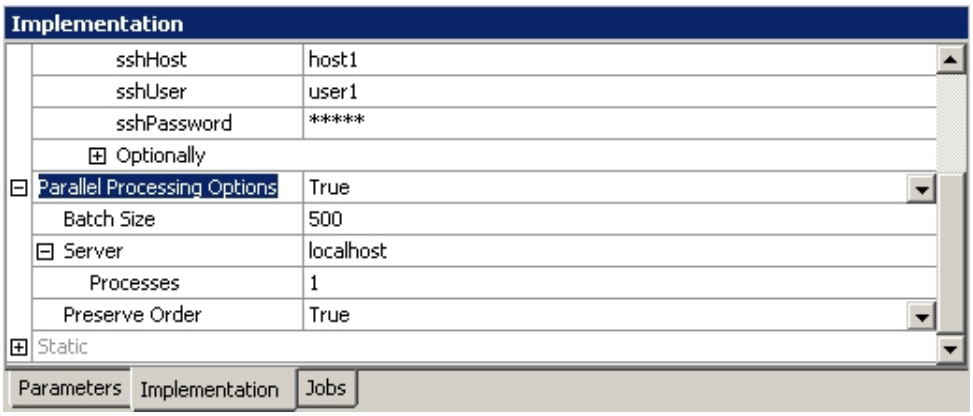

Figure 3.4: The options for tuning parallel processing are found in the Implementation tab

to read the Pipeline Pilot documentation about parallel processing to understand more fully how it works.

Note: to set up large parallel processing jobs, you will need Administrator rights to change one setting.

3.5.1 The Default Set-Up: Maintaining the Pipeline Effect

By default, the components are set to run in parallel on the *localhost* (the Pipeline Pilot server) with only one process - in essence not a parallel calculation at all! These are the default values, below we give an explanation for these settings.

- *on PP Server* method of connection to the external *FTrees* installation (*FTrees* is installed on the Pipeline Pilotserver).
- <Parallel Processing Options> is set to *True*.
- <Server> set to *localhost*
- <Processes> 1

The idea behind using these settings is to enhance the pipeline effect for the *FTrees* components, not to run a true parallel calculation. By choosing these settings, Pipeline Pilot is forced to collect the incoming pipeline molecules into sets. As soon as one set is complete (a set contains the number of molecules set by the parameter <Batch Size>) Pipeline Pilot starts the calculations for the component. Hence, the *FTrees* components are already running while the pipeline continues to deliver molecules, saving time. As soon as all molecules have arrived, the pipeline continues. Note that the <Batch Size> parameter for both components is set more with this idea in mind, rather than for a true parallel calculation.

If the option <Parallel Processing Options> is set to *False*, all molecules must arrive at the *FTrees* component before a calcuation is started. This means a little performance may be gained by avoiding the overhead required to split the pipeline into batches, however, the

FTrees component may only start when it has all molecules and the pipeline can only begin again once the *FTrees* component is finished.

Even if you do not have a multi-processor server, or do not wish to calculate in parallel, we recommend you still keep these default options to enhance the pipeline effect as explained above.

3.5.2 Running in Parallel on the Pipeline Pilot Server

You may have a multi-processor machine as your Pipeline Pilot server. If you also have the appropriate number of Pipeline Pilot and *FTrees* licenses the simplest way to start a parallel calculation is to raise the number of processes to the number of processors of the machine.

You could also have more than one Pipeline Pilot server available in your network. If so, you can enter a list of the server names at the parameter <Server>. Below that, for the parameter <Processes> enter a list of the number of processes each server should receive. The lists are both comma separated and must be in corresponding order. Remember to adjust also the <Batch Size> accordingly.

Note: the path to the external installation of *FTrees* must be same on all servers!

3.5.3 Running in Parallel on a Remote Linux Cluster

We have developed an implementation in the components whereby a large cluster can be incorporated to run *FTrees* jobs, without them having to be Pipeline Pilot servers. However, it must be a Linux cluster and the components must use the ssh method. You must also have enough *FTrees* licenses available to the cluster.

The settings in the Implementation tab must be made as for running the ssh method, with two important changes. Instead of one <Run FTrees -> via SSH -> Host>, enter a comma separated list of the host names in the Linux cluster. Then, you must choose how many batchs in total you want the job to be split into and enter this total in the <Processes> parameter. Remember to change the <Batch Size> to fit with the number of processes.

Figure 3.5 may help clarify how the method works.

The Pipeline Pilot server will run with the number of processes given in <Processes>. However, in this case the processes the server receives are not the *FTrees* calculations themselves but instructions for starting the ssh jobs. The server iterates through its 10 jobs each time spawning an ssh job on a Linux host. Beware that the Pipeline Pilot server does not know how many processors the Linux hosts have so make sure you choose the number of <Processes> to fit the number of Linux hosts and their number of processors respectively – be careful not to overload the Linux hosts.

Figure 3.6 shows the detailed settings required to set up the calculation as in figure 3.5.

As you may already have realised, you could enter more than one Pipeline Pilot server at the <Server> parameter (along with another entry for number of <Processes> as a comma separated list) to execute a doubly parallel calculation!

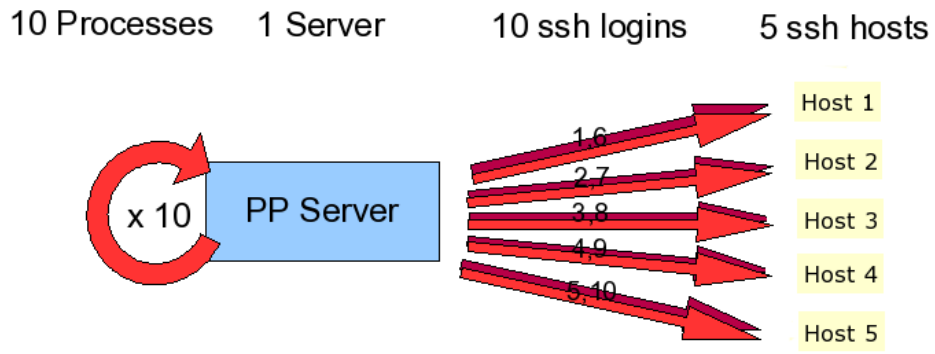

Figure 3.5: The method of parallel processing on Linux clusters in the *FTrees* components. The Pipeline Pilot server is given 10 processes. The 10 processes start an ssh job respectively, distributed amongst the ssh hosts.

Note: to get this method to work you will need to change the maximum number of processes per Pipeline Pilot server. The Pipeline Pilot Client will let you enter any number for the <Processes> parameter and does not warn you if this number is above the maximum.

Changing the maximum number requires Administrator rights! The number of processes per Pipeline Pilot server is usually restricted to the number of processors of the server. You must override this maximum to be able to set the number of processes you want for your parallel calculation. In the above example, the maximum must be set to 10 or more. Take the following steps:

- Go to the 'Scitegic Server Home Page', for example, via the Help menu in your Pipeline Pilot client.
- Click on 'Pipeline Pilot Administration Portal' and log in with the Administrator user name and password.
- In the last field of the table ('Maximum number of simultaneous parallel processing subprotocols allowed') change the value, click 'Save' and log out again.

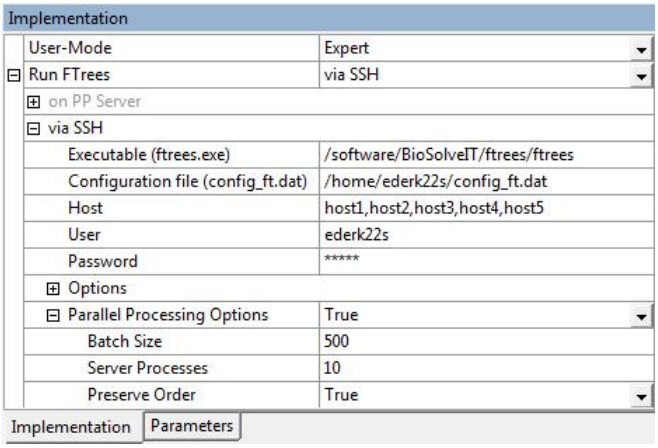

Figure 3.6: The options for parallel processing for the example shown in figure 3.5 would look similar to these

3.5.4 Example Scenarios and Required Settings

ssh/ops	*ops*
Parallel Processing Options	False
Number of PP servers in list	1
Number of ssh Hosts in list	-
Number of Processes	-
Behavior	The calculation will run as one complete job on the Pipeline Pilot server
ssh/ops	*ssh*
Parallel Processing Options	False
Number of PP servers in list	-
Number of ssh Hosts in list	1
Number of Processes	-
Behavior	The calculation will run as one complete job on the ssh host
ssh/ops	*ops*
Parallel Processing Options	True
Number of PP servers in list	1
Number of ssh Hosts in list	-
Number of Processes	1
Behavior	(Default) Induces the 'pipeline' effect: the job will run in chunks on the server processor(s)
ssh/ops	*ops*
Parallel Processing Options	True
Number of PP servers in list	1
Number of ssh Hosts in list	-
Number of Processes	>1
Behavior	A true parallel effect: the job will be run in chunks on the server processors
ssh/ops	*ops*
Parallel Processing Options	True
Number of PP servers in list	4 (each with two processors)
Number of ssh Hosts in list	-
Number of Processes	2,2,2,2
Behavior	A true parallel effect: the job will run in chunks in parallel split across 8 processors
ssh/ops	*ssh*
Parallel Processing Options	True
Number of PP servers in list	1
Number of ssh Hosts in list	5 (each with 2 processors)
Number of Processes	10
Behavior	A true parallel effect: the job will run in chunks in parallel split across 10 processors

3.6 Uninstallation

Use the *FTrees* installer component to uninstall previous *FTrees* versions by switching Parameter "Uninstall Previous Versions" to "True".

Use BioSolveIT Uninstaller component to uninstall all *FTrees* versions.

Note that this kind of deinstallation only removes *FTrees* versions installed by *FTrees* installer component or BioSolveIT web installer component.

4

Trouble Shooting

4.1 Problems connecting to the external installation

Other most commonly seen problems with *FTrees* in Pipeline Pilot is with the connection to the external *FTrees* installation. For one thing, *FTrees* itself must be correctly installed on the system independently of Pipeline Pilot – it is essential first to make sure this is the case (especially to make sure that *FTrees* can locate the licenses). Once *FTrees* runs fine on your system, the remaining key task is to make sure the paths to the executable and configuration file are correct within the Pipeline Pilot components.

The simplest test to check whether *FTrees* is running OK in Pipeline Pilot is to set up the protocol shown in section 2.3.1 with just 10 molecules or so. If something is amiss with the connection to the external installation, you will see an error message box pop up. Check the paths to the executable or to the configuration file. If it seems *FTrees* could be started but not run, the problem almost always lies either with the path to the configuration file or with licenses not being found. If *FTrees* runs OK independently from Pipeline Pilot then it is likely to be the path to the configuration.

When the error messages pop up, they may contain an FTreesError in the error message box, as in figure 4.1.

Go to the 'Jobs' tab below the Protocol workspace and check under the last run job for a file called 'FTreesComponent Debug' – as in figure 4.2. Clicking on the link brings up HTML report with input and output data in a browser.

A correctly started *FTrees* job outputs the following header: if there is a problem you will see some of this header and the point where the problem occurs:

```
                F T R E E S
      Feature-based molecular similarity finder
  Version:  2.0.2  (11.08.08)
  Modules: [PVM] [FFS]
 written by:  Matthias Rarey, Marc Zimmermann,
```

```
       /               Sally Hindle, Robert Fischer             /
      /                                                         /
     / copyright by: BioSolve IT GmbH, FhG -SCAI               /
    /                Sankt Augustin, Germany                  /
   / for further information mail ftrees-info@biosolveit.de /
  /_____________________________________________________/
 Additional copyright notes:
   getline library: (C) 1993 by Chris Thewalt
   PVM library: (C) 1997 by University of Tennessee, Knoxville TN
 >> FTrees configuration file '/software/BioSolveIT/FTrees/ftrees2.0.1/
    config_ft.dat' loaded.

 >> Licensed modules: FTrees [PVM] [FFS]
 >> PVM status: no pvm daemon, running sequential.
 >> Scripts are executed in sequential mode; start PVM for parallel mode.
 >> SETTINGS = '/software/BioSolveIT/ftrees2.0.1/static_data/ftrees_settings.dat'
    loaded.
 >> CHEMPAR  = '/software/BioSolveIT/ftrees2.0.1/static_data/chempar.dat' loaded.
 >> CONTACT  = '/software/BioSolveIT/ftrees2.0.1/static_data/contact_ft.dat' loaded.
 >> TRANSFORM = '/software/BioSolveIT/ftrees2.0.1/static_data/transform.dat' loaded.
 >> GRAPHIC  = '/software/BioSolveIT/ftrees2.0.1/static_data/graphic_ft.dat' loaded.
...
```

4.2 Problems using the *ssh* Method

You may also experience problems using the *ssh* login, for example, the user name is unknown or the host is not found.

4.3 Further help and BioSolveIT PDF Reporter

More complicated errors may arise during the running of *FTrees*. Again though, the errors will be collected and as much information shown as possible. If you are familiar with *FTrees* you may want to take a look at all the output of the job yourself to see if you can recognize the problem. In this case, you can look in the temporary folders Pipeline Pilot sets up internally to find the output, or if you are working with the *ssh* method, set the parameter <Run FTrees -> via SSH -> Options -> Delete Results> to *False* so that you may then find the files retained on the *ssh* host: these will be in the directory set under the ssh parameter <Run FTrees -> via SSH -> Options -> Temp Path> (see the help text associated with this parameter to find its default value) – essentially a cryptically named folder whose name begins with the date and time of the job!

If you still do not know what is causing the errors, write down as much information as possible relating to your installation scheme. You can also create a PDF Report unsing BioSolveIT PDF Reporter component which summarizes installation data into one pdf file. Send all the information to:

support@biosolveit.de

Figure 4.1: An error box reporting that the *FTrees* exe exited with error.

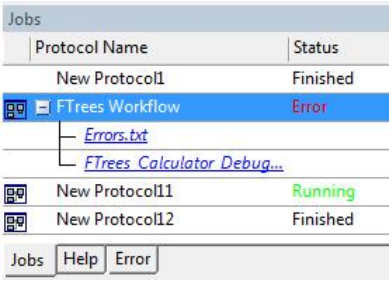

Figure 4.2: The full error report can be found in the 'Jobs' tab.

5

Tips and Tricks

5.1 Other Significant Parameters in the *FTrees* Components

For detailed documentation of all parameters for all components, refer to the documentation you find in the *Help* area of the Pipeline Pilot window.

We list here particularly interesting parameters: those that greatly influence the protocol or change the outcome of calculations, or those that may help you understand what is happening in the components.

5.1.1 <Molecule Initialization>

FTrees Calculator : Parameters : <FTrees Options -> Library Molecule Initialization>
FTrees Similarity : Parameters : <FTrees Options -> Query Molecule Initialization>

FTrees itself processes the molecules that it reads to for example, check the protonation states, check the charges, aromacity and set SYBYL-like atom types that it needs. This is called molecule initialization and varies slightly according to the file type of the molecule file.

By default in Pipeline Pilot the molecules are always passed to *FTrees* in MOL2 format as this retains the most information about the molecule, thus retaining any manipulation relating to chemistry carried out in Pipeline Pilot since it was read from file. In principle then, if you are sure that your molecules are properly prepared either in the pipeline or already in the molecule file, then you could try switching this parameter to *False* – this makes the Calculator component run much faster.

However, the reliability of the results cannot be guaranteed. The MOL2 sent to the *FTrees* calculation relies a little on the Pipeline Pilot MOL2 writer, which may deviate in its definition of some atom types to *FTrees*. Also, in *FTrees*, it is not usually relevant exactly *how* the molecules are prepared for *FTrees* as long as they all pass through the same prepartion process. It is unlikely that this will be consistent across the queries and pipeline molecules if the feature trees were calculated for the pipeline with <Molecule Initialization> *False*.

We recommend leaving <Molecule Initialization> set to *True* for both components to maximize consistency.

5.1.2 `.ftdb` Compression

FTrees Writer/Reader `.ftdb` compression

The file written by the *FTrees* Writer must be of the type `.ftdb`. True, but not entirely! If you enter the extension `.ftdb.zip` or `.ftdb.gz` the Writer will automatically compress the output file. Similarly with the Reader: you may enter files with the extensions `.ftdb.zip` or `.ftdb.gz` and the Reader will automatically decompress them.

5.1.3 Fail ports for *FTrees* Calculator and Simlarity

The Fail ports for the *FTrees* Calculator outputs molecules for which no feature tree could be calculated. This may seem logical for Pipeline Pilot but it is an additional feature for *FTrees* that does not exist in the original tool!

Analog to this, the *FTrees* Similarity component outputs molecules for which no similarity was calculated to the Fail port, again a unique feature. However, these are output amongst other 'Fail' molecules for example for which no feature tree was found, or for which the simlarity was below the set threshold using <SimilarityThreshold>.

5.1.4 <File Format>

FTrees Similarity : Implementation : <FTrees Configuration -> Query File Format> *FTrees* Calculator : Implementation : <FTrees Configuration -> Library File Format>

This parameter becomes relevant when more than one query were used in the *FTrees* Similarity component (more than one query in the query input file). *FTrees* Similarity calculates one similarity score per pipeline molecule per query. All scores per pipeline molecule are annotated to the molecule data record. When there is more than one score to be added, how best to arrange this as a data record Property? With this parameter you can choose yourself how.

Select <ColumnPerQuery> to add a separate Property to the data record per query. When viewed in a table viewer, you will see a new column of data for each query labeled with 'BSIT_sim_<query name>'. This is useful if you want to manipulate the molecules further down the pipeline based on the score for an individual query.

On the other hand, you may be more interested in manipulating molecules based on any of the similarity scores. For this, the option <SimilarityColumn> will be more useful. Here, all scores are written as a Property array and will appear as one column when viewed in the table. The Generic components 'Unmerge Data' and 'Merge Data' will also be useful in this case.

5.1.5 <Has Same File System>

*FTrees*Calculator/Similarity : Implementation : <Run FTrees-> via SSH -> Options -> Has Same File System>

Normally, for an *ssh* job, Pipeline Pilot must first copy all the data required by FTrees to the <Run FTrees -> via SSH -> Host> using scp. This is time consuming. It is possible that the Pipeline Pilot Server and <Run FTrees -> via SSH -> Host> actually share the same file system rendering the scp process unnecessary. Select True if the Pipeline Pilot Server and <Run FTrees -> via SSH -> Host> share the same File System - no copying of data is necessary. Selecting False means Pipeline Pilot copies all the data to and back from the <Run FTrees -> via SSH -> Host> – this is just a little slower but will always still work. Leave the parameter set to False if you are uncertain!

5.2 Writing a standard *FTrees* .fdf file

The Pipeline Pilot Text Writer component can be used in a protocol after the *FTrees* Calculator component to write feature trees to a standard *FTrees* file .fdf to use in standalone *FTrees*. To do this, set up the Text Writer component as shown in figure 5.1.

5.3 Reading a standard *FTrees* .fdf file for the Similarity Component

The Pipeline Pilot Text Reader component can be used before the *FTrees* Similarity component to read in feature trees from a standard *FTrees* file .fdf to the pipeline and calculate similarities for them.
Note: the pipeline will contain plain data records and not molecule data records, there will be no molecules associated with the similarity values in this case.
To do this, set up the Text Reader component as shown in figure 5.2.

5.4 Accessing Other Domains within Pipeline Pilot

Often in house data or even your own working data are accessible from a windows computer via a domain (a path starting for example 'z:\...' or '\\...') which you cannot find from within Pipeline Pilot. That means you must first literally transfer the data to the Pipeline Pilot Server itself.

If you are using a Linux Pipeline Pilot server, this hint does not apply.

To get around this problem and make the Pipeline Pilot working environment much more flexible you can allow users access to domains – you need Administrator rights to be able to do this! Also check first that you should change these settings as they may have already been set to fit the current environment.

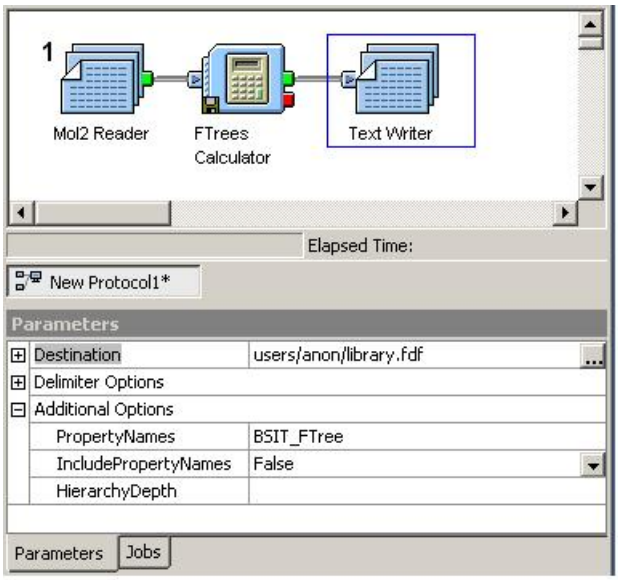

Figure 5.1: *FTrees* `.fdf` files can be written with the Text Writer

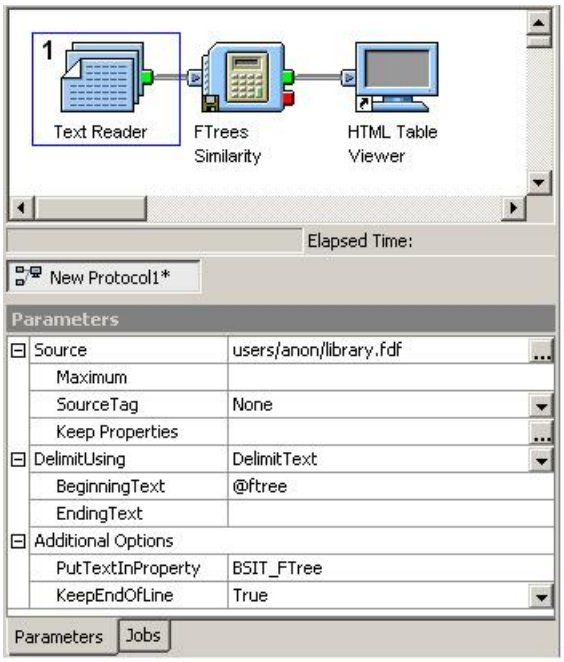

Figure 5.2: *FTrees* `.fdf` files can be read with the Text Reader

- Go to the 'Scitegic Server Home Page', for example, via the Help menu in your Pipeline Pilot client.
- Click on 'Pipeline Pilot Administration Portal' and log in with the Administrator user name and password.
- In the Security tab go to Authentification.
- For the 'Authentification Method' choose 'DOMAIN' and a set of parameters will appear.
- Enter the domain name in the field 'Domain' and choose 'Full' for 'Impersonation'
- Choose 'DOMAIN' for 'Retrieve Groups' and leave 'Limit access to listed domains' set to 'No'
- click 'Save' and log out again.

After you have done this you will need to enter your domain login details when you start the Pipeline Pilot Client.

Bibliography

[1] M. Rarey and J.S. Dixon. Feature trees: A new molecular similarity measure based on tree matching. *Journal of Computer-Aided Molecular Design*, 12:471–490, 1998. 7

[2] M. Rarey and M. Stahl. Similarity searching in large combinatorial chemistry spaces. *Journal of Computer-Aided Molecular Design*, 15:497–520, 2001. 2